

**IN THE CLAIMS:**

1. (Currently Amended) A method for managing communications between requester processes and server processes in a data processing network, including:
  - creating a set of dispatcher processes, each have a unique dispatcher process identifier;
  - associating each of a set of requester processes, which communicate with a server process via a common interpreter process having a common process identifier, with a unique dispatcher process of said set of dispatcher processes, wherein each of the set of requester processes executes within the common interpreter process having the common process identifier and wherein each of the set of requester processes shares the common process identifier;
  - for requests sent from any of said set of requester processes via said common interpreter process to a server process which identifies each of said set of requester processes using the unique dispatcher process identifier, routing said requests via the associated dispatcher process;
  - at the respective dispatcher process, attaching the unique dispatcher process identifier to the request and then forwarding the request to the server process; and
  - responsive to receipt by the dispatcher process of a reply to said request, forwarding the reply to the associated requester process via the common interpreter process.
2. (Previously Presented) A method according to claim 1, wherein the common interpreter process via which each of said set of requester processes associated with the unique dispatcher process communicates is a Java Virtual Machine.
3. (Original) A method according to claim 2, wherein the set of requester processes comprise Web Browsers which communicate with a server process via respective Servlet threads running within a JVM of a Web Server or Web application server.

4. (Currently Amended) A computer program product comprising program code recorded on a machine readable recording medium, the program code including instructions for, when executed, controlling the operating of a data processing apparatus to implement a method for managing communications between requester processes and server processes in a data processing network, the method including:

creating a set of dispatcher processes, each having a unique dispatcher process identifier;

associating each of a set of requester processes, which communicate with a server process via a common interpreter process having a common process identifier, with a unique dispatcher process of said set of dispatcher processes, wherein each of the set of requester processes executes within the common interpreter process having the common process identifier and wherein each of the set of requester processes shares the common process identifier;

for requests sent from any of said set of requester processes via said common interpreter process to a server process which identifies each of said set of requester processes using the unique dispatcher process identifier, routing said requests via the associated dispatcher process;

at the respective dispatcher process, attaching the unique dispatcher process identifier to the request and then forwarding the request to the server process; and

responsive to receipt by the dispatcher process of a reply to said request, forwarding the reply to the associated requester process.

5. (Currently Amended) A data processing apparatus, including:

a server process which uses identifiers to distinguish between requests received from different client processes;

means for creating a set of dispatcher processes, each having a unique dispatcher process identifier;

means for associating each of a set of requester processes, which communicate with the server process via a common interpreter process having a common process identifier, with a unique dispatcher process of said set of dispatcher processes, wherein each of the set of requester processes executes within the common interpreter process

having the common process identifier and wherein each of the set of requester processes shares the common process identifier;

means for routing requests from a requester process, comprising requests sent to the server process from any of said set of requester processes via the common interpreter process, via the respective associated dispatcher process;

means associated with the respective dispatcher process for attaching the unique said respective dispatcher process identifier to the request and then forwarding the request to the server process; and

means responsive to receipt by said respective dispatcher process of a reply to said request, for forwarding the reply to the associated requester process.

6. (New) The method of claim 1, wherein associating each of a set of requester processes with a unique dispatcher process of said set of dispatcher processes includes invoking a service module to request a unique dispatcher process for each of the set of requester processes.

7 (New) The method of claim 6, wherein the service module creates a unique dispatcher process for each of the set of requester processes and assigns each unique dispatcher process to each of the set of requester processes.

8. (New) The method of claim 7, wherein the service module maintains a table of relationships between each of the requester processes and assigned unique dispatcher process.

9. (New) The method of claim 6, wherein the service module is invoked by calling a register method implemented by each of the set of requester processes.

10. (New) The method of claim 1, wherein routing said requests via the associated dispatcher process includes invoking a send method implemented by a native dynamically linked library.

11. (New) The data processing apparatus of claim 5, wherein the means for associating each of a set of requester processes with a unique dispatcher process of said set of dispatcher processes includes means for invoking a service module to request a unique dispatcher process for each of the set of requester processes.
12. (New) The data processing apparatus of claim 11, wherein the service module includes means for creating a unique dispatcher process for each of the set of requester processes and means for assigning each unique dispatcher process to each of the set of requester processes.
13. (New) The data processing apparatus of claim 12, wherein the service module includes means for maintaining a table of relationships between each of the requester processes and assigned unique dispatcher process.
14. (New) The data processing apparatus of claim 11, wherein means for invoking a service module includes means for calling a register method implemented by each of the set of requester processes.
15. (New) The data processing apparatus of claim 5, wherein means for routing said requests via the associated dispatcher process includes means for invoking a send method implemented by a native dynamically linked library.